This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. **(Currently Amended)** An apparatus that determines allocations in a business operation to maximize profit on a computer system, comprising:

a memory; and

a processor that accesses the memory to retrieve computer-executable instructions to perform:

collecting profit data for a plurality of classes in the business operation, each class including an allocation having a cost function, the allocations being constrained by a total floor area, each class corresponding to a department of the business operation, and each allocation belonging to the group consisting of physical allocations and economic allocations;

determining profit functions for the allocations from the profit data by:

determining demand distributions for the allocations from the profit data;

determining a spatial allotment for each said department; and

determining each profit function from a corresponding demand distribution for the spatial allotment of each said department;

formulating a Multiple Choice Knapsack Problem to maximize profit ~~from~~ based one the profit functions, the cost functions, and a cost constraint; ~~and~~

~~solving~~ running the Multiple Choice Knapsack Problem to determine values for the allocations, by utilizing ~~a recursive function that rewrites values into~~ a solution vector~~, the solution vector~~ holding allocation values, and a ~~determined from recursive running of the~~ recursive function that rewrites allocation values into the solution vector by recursively running

for possible allocations from 0 up to a predetermined maximum allocation for each class, wherein the recursive function evaluates possible allocations for each class in a sequentially increasing order to improve local caching performance; and

selecting the allocation values in the solution vector that maximize profit.

2. **(Cancelled)**

3. **(Previously Presented)** The apparatus according to claim 1, wherein each demand distribution includes a Poisson model.

4. **(Previously Presented)** The apparatus according to claim 1, wherein each demand distribution includes a Markov model.

5. **(Previously Presented)** The apparatus according to claim 1, wherein each demand distribution includes a normal distribution model.

6. **(Previously Presented)** The apparatus according to claim 1, wherein the allocations include spatial allotments.

7. **(Previously Presented)** The apparatus according to claim 1, wherein the allocations include monetary allotments.

8. **(Previously Presented)** The apparatus according to claim 1, wherein the cost constraint is a greater-than-or-equal-to inequality constraint.

9. **(Previously Presented)** The apparatus according to claim 1, wherein the cost constraint is an equality constraint.

10. **(Previously Presented)** The apparatus according to claim 1, wherein the cost constraint is a less-than-or-equal-to inequality constraint.

11. **(Currently Amended)** An apparatus that determines physical allocations in a business operation to maximize profit on a computer system, comprising:

a memory; and

a processor that accesses the memory to retrieve computer-executable instructions to perform:

collecting profit data for a plurality of classes in the business operation, each class including a physical allocation having a cost function, the physical allocations being constrained by a total floor area, each class corresponding to a department of the business operation;

determining profit functions for the physical allocations from the profit data by:

determining demand distributions for the allocations from the profit data;

determining a spatial allotment for each said department; and

determining each profit function from a corresponding demand distribution for the spatial allotment of each said department;

formulating a Multiple-Choice Knapsack Problem to maximize profit ~~from~~ based on the profit functions, the cost functions, and a cost constraint; and

~~solving~~ running the Multiple Choice Knapsack Problem to determine values for the physical allocations, by utilizing a solution vector holding physical allocation values, and a recursive function that rewrites values into ~~a~~ the solution vector~~, the solution vector holding values determined from recursive running of the recursive function~~ by recursively running for possible physical allocations from 0 up to a predetermined maximum physical allocation for each class, wherein the recursive function evaluates possible physical allocations for each class in a sequentially increasing order to improve local caching performance; and

selecting the physical allocation values in the solution vector that maximize profit.

12. **(Cancelled)**

13. **(Previously Presented)** The apparatus according to claim 11, wherein each demand distribution includes a Poisson model.

14. **(Previously Presented)** The apparatus according to claim 11, wherein each demand distribution includes a Markov model.

15. **(Previously Presented)** The apparatus according to claim 11, wherein each demand distribution includes a normal distribution model.

16. **(Previously Presented)** The apparatus according to claim 11, wherein the physical allocations include spatial allotments for the classes.

17. **(Previously Presented)** The apparatus according to claim 16, wherein the spatial allotments include widths for the classes and the cost constraint is a width constraint.

18. **(Previously Presented)** The apparatus according to claim 16, wherein the spatial allotments include advertising spaces for the classes and the cost constraint is an advertising space constraint.

19. **(Previously Presented)** The apparatus according to claim 16, wherein the spatial allotments include catalog spaces for the classes and the cost constraint is a catalog space constraint.

20. **(Previously Presented)** The apparatus according to claim 16, wherein the spatial allotments include floor spaces for the classes and the cost constraint is a floor space constraint.

21. **(Previously Presented)** The apparatus according to claim 11, wherein the cost constraint is a greater-than-or-equal-to inequality constraint.

22. **(Previously Presented)** The apparatus according to claim 11, wherein the cost constraint is an equality constraint.

23. **(Previously Presented)** The apparatus according to claim 11, wherein the cost constraint is a less-than-or-equal-to inequality constraint.

24. **(Currently Amended)** An apparatus that determines economic allocations in a business operation to maximize profit on a computer system, comprising:

a memory; and

a processor that accesses the memory to retrieve computer-executable instructions to perform:

collecting profit data for a plurality of classes in the business operation, each class including an economic allocation having a cost function, the economic allocations being constrained by a total floor area, each class corresponding to a department of the business operation;

determining profit functions for the economic allocations from the profit data by:

determining demand distributions for the allocations from the profit data;

determining a spatial allotment for each said department; and

determining each profit function from a corresponding demand distribution for the spatial allotment of each said department;

formulating a Multiple Choice Knapsack Problem to maximize profit based on ~~from~~ the profit functions, the cost functions, and a cost constraint; and

~~solving~~ running the Multiple Choice Knapsack Problem to determine values for the economic allocations, by utilizing a solution vector holding economic allocation values, and a recursive function that rewrites values into ~~a~~ the solution vector~~, the solution vector holding~~

~~values determined from recursive running of the recursive function~~ by recursively running for possible economic allocations from 0 up to a predetermined maximum economic allocation for each class, wherein the recursive function evaluates possible economic allocations for each class in a sequentially increasing order to improve local caching performance; and

selecting the economic allocation values in the solution vector that maximize profit.

.

25. **(Cancelled)**

26. **(Previously Presented)** The apparatus according to claim 24, wherein each demand distribution includes a Poisson model.

27. **(Previously Presented)** The apparatus according to claim 24, wherein each demand distribution includes a Markov model.

28. **(Previously Presented)** The apparatus according to claim 24, wherein each demand distribution includes a normal distribution model.

29. **(Previously Presented)** The apparatus according to claim 24, wherein the economic allocations include monetary allotments for the classes.

30. **(Previously Presented)** The apparatus according to claim 29, wherein the cost constraint is a monetary constraint.

31. **(Previously Presented)** The apparatus according to claim 24, wherein the cost constraint is a greater-than-or-equal-to inequality constraint.

32. **(Previously Presented)** The apparatus according to claim 24, wherein the cost constraint is an equality constraint.

33. **(Previously Presented)** The apparatus according to claim 24, wherein the cost constraint is a less-than-or-equal-to inequality constraint.

34. **(Currently Amended)** A system for determining allocations in a business operation to maximize profit, comprising:

a data unit, the data unit having a memory that includes profit data for a plurality of classes in the business operation, each class including an allocation having a cost function that is stored in the memory, and the memory also including a cost constraint, the allocations being constrained by a total floor area, each class corresponding to a department of the business operation;

a profit-model unit, the profit-model unit being connected to the data unit, and the profit-model unit including executable instructions for determining profit functions for the allocations from the profit data, wherein determining the profit functions includes:

determining demand distributions for the allocations from the profit data;

determining a spatial allotment for each said department; and

determining each profit function from a corresponding demand distribution for the spatial allotment of each said department; and

an optimization-engine-unit, the optimization-engine unit being connected to the data unit and the profit-model unit, the optimization-engine unit including executable instructions for formulating a Multiple Choice Knapsack Problem to maximize profit ~~from~~ based on the profit functions, the cost functions, and the cost constraint, for creating a solution vector holding allocation values, and for ~~solving~~ running the Multiple Choice Knapsack Problem to determine values for the allocations, by utilizing a recursive function that rewrites values into the solution vector~~, the solution vector holding values determined from recursive running of the recursive function~~ by recursively running for possible physical from 0 up to a predetermined maximum

allocation for each class, wherein the recursive function evaluates possible allocations for each class in a sequentially increasing order to improve local caching performance; and

wherein the optimization-engine unit including executable instructions for selecting the allocation values in the solution vector that maximize profit.

35. **(Cancelled)**

36. **(Previously Presented)** A system according to claim 34, wherein each demand distribution includes a Poisson model.

37. **(Previously Presented)** A system according to claim 34, wherein each demand distribution includes a Markov model.

38. **(Previously Presented)** A system according to claim 34, wherein each demand distribution includes a normal distribution model.

39. **(Original)** A system according to claim 34, wherein the allocations include spatial allocations.

40. **(Original)** A system according to claim 34, wherein the allocations include economic allocations.

41. **(Original)** A system according to claim 34, wherein the cost constraint is a greater-than-or-equal-to inequality constraint.

42. **(Original)** A system according to claim 34, wherein the cost constraint is an equality constraint.

43. **(Original)** A system according to claim 34, wherein the cost constraint is a less-than-or-equal-to inequality constraint.

44. **(Currently Amended)** Computer-readable media tangibly embodying a program for determining allocations in a business operation to maximize profit, the program including executable instructions for:

collecting profit data for a plurality of classes in the business operation, each class including an allocation having a cost function, the allocations being constrained by a total floor area, each class corresponding to a department of the business operation;

determining profit functions for the allocations from the profit data by:

determining demand distributions for the allocations from the profit data;

determining a spatial allotment for each said department; and

determining each profit function from a corresponding demand distribution for the spatial allotment of each said department;

formulating a Multiple Choice Knapsack Problem to maximize profit ~~from~~ based on from the profit functions, the cost functions, and a cost constraint; and

solving the Multiple Choice Knapsack Problem to determine values for the allocations, by utilizing a solution vector holding allocation values, and a recursive function that rewrites values into ~~a~~ the solution vector~~, the solution vector holding values determined from recursive running of the recursive function~~ by recursively running for possible allocations from 0 up to a predetermined maximum allocation for each class, wherein the recursive function evaluates possible allocations for each class in a sequentially increasing order to improve local caching performance; and

selecting the allocation values in the solution vector that maximize profit.

.

45. **(Cancelled)**

46. **(Previously Presented)** Computer-readable media as claimed in claim 44, wherein each demand distribution includes a Poisson model.

47. **(Previously Presented)** Computer-readable media as claimed in claim 44, wherein each demand distribution includes a Markov model.

48-53 **(Cancelled)**

54. **(Previously Presented)** The apparatus of claim 1, wherein determining demand distributions for the allocations from the profit data comprises:

modeling the demand distributions with corresponding probabilistic functions.

55. **(Cancelled)**

56. **(Previously Presented)** The apparatus of claim 1, wherein a size of the solution vector is substantially equal to a value of the cost constraint.

57. **(Previously Presented)** The apparatus of claim 1, further including computer-executable instructions to perform:

before solving the Multiple Choice Knapsack Problem, ordering the plurality of classes by cost function.

58. **(Previously Presented)** The apparatus of claim 1, further including computer-executable instructions to perform:

before solving the Multiple Choice Knapsack Problem, removing a first item from a class if the item is dominated by a second item in the class, wherein the second item dominates the first item if the first and second item have substantially similar cost functions, but the second item has a higher profit function.

59. **(Previously Presented)** The apparatus of claim 24, wherein a size of the solution vector is substantially equal to a value of the cost constraint.

60. **(Previously Presented)** The apparatus of claim 24, further including computer-executable instructions to perform:

before solving the Multiple Choice Knapsack Problem, ordering the plurality of classes by cost function.

61. **(Previously Presented)** The apparatus of claim 24, further including computer-executable instructions to perform:

before solving the Multiple Choice Knapsack Problem, removing a first item from a class if the item is dominated by a second item in the class, wherein the second item dominates the first item if the first and second item have substantially similar cost functions, but the second item has a higher profit function.